Shift Register Sequences

Based on S. W. Golomb, *Shift Register Sequences*,
Aegean Park Press (1982, Revised Edition)

Prof. Michael Mascagni

Department of Computer Science
Department of Mathematics
Department of Scientific Computing
Graduate Program in Molecular Biophysics
Florida State University, Tallahassee, FL 32306 **USA**
E-mail: mascagni@fsu.edu
URL: http://www.cs.fsu.edu/∼mascagni

Overview

Introduction & Motivation

Linear Feedback Shift Registers (§2)

Properties of m-Sequences (§3–4)

Cycle Structure of Shift Registers (§2)

Cross-Correlation & Gold Sequences (§5–6)

Non-Linear Shift Registers & Applications (§7–8)

Summary & Further Reading

What Is a Shift Register?

Definition:

A **shift register of length** n is a clocked device consisting of n storage cells (stages), each holding one symbol from a finite alphabet $\mathcal{A}$. At each clock tick:

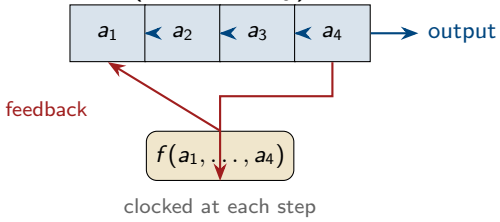
1. The content of each cell shifts *one position to the right*.
2. The new leftmost cell is filled by a **feedback function** $f(a_1, a_2, \ldots, a_n)$ applied to the current state.

Output: the sequence of symbols exiting from the rightmost cell.

Binary case (most common)

$\mathcal{A} = \{0, 1\} = \mathrm{GF}(2)$.
The feedback function $f : \mathrm{GF}(2)^n \to \mathrm{GF}(2)$ is a Boolean function.

Schematic ($n = 4$, binary):



State: the n-tuple $(a_1, a_2, \ldots, a_n) \in \mathrm{GF}(2)^n$.
There are 2^n possible states.

Why Shift Registers Matter for RNG

Shift registers are everywhere:

- **Pseudorandom number generation** — LFSRs produce long, statistically excellent sequences at very low hardware cost
- **Spread-spectrum communications** — PN sequences used for synchronization and channel coding
- **Cryptography** — stream cipher key-stream generators
- **Error-correcting codes** — BCH codes, Reed-Solomon codes
- **VLSI testing** — built-in self-test (BIST)
- **Radar ranging** — pulse compression via m-sequences

Golomb's central question

Under what conditions does a shift register of length n produce a **single long cycle** of period $2^n - 1$, and what are the statistical properties of the output sequence?

This course's focus

- **Linear** feedback shift registers (LFSRs)
- **Maximal-length** sequences (m-sequences)
- **Golomb's three randomness postulates**
- **Correlation** properties

Linear Feedback Shift Registers (LFSRs)

The linearity restriction:

Restrict the feedback function to be **linear over** $\mathrm{GF}(2)$:

$$a_{n+k} = c_1 a_{n+k-1} \oplus c_2 a_{n+k-2} \oplus \cdots \oplus c_n a_k, \quad c_i \in \mathrm{GF}(2)$$

where $\oplus$ denotes addition mod 2 (XOR).

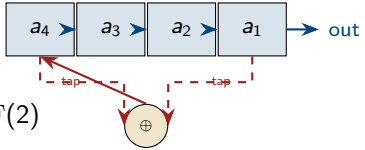
Characteristic polynomial:

$$\boxed{f(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_n x^n \in \mathrm{GF}(2)[x]}$$

Example ($n = 4$):
$f(x) = 1 + x + x^4$ (taps at positions 1 and 4)

$$a_{k+4} = a_{k+3} \oplus a_k$$

LFSR with $f(x) = 1 + x + x^4$:



Key fact

The all-zero state **0** maps to itself and is **excluded** from useful sequences. At most $2^n - 1$ non-zero states exist.

Period, Primitive Polynomials, and m-Sequences

Period of an LFSR:
The *period* T of the output sequence divides $2^n - 1$.
T is determined by the characteristic polynomial $f(x)$.

Key definitions

- $f(x)$ is **irreducible** over $\mathrm{GF}(2)$ if it has no non-trivial factors in $\mathrm{GF}(2)[x]$
- $f(x)$ is **primitive** if it is irreducible and its roots have multiplicative order $2^n - 1$ in $\mathrm{GF}(2^n)$
- An LFSR with a primitive $f(x)$ produces a **maximal-length sequence (m-sequence)** of period $T = 2^n - 1$

Primitive polynomials for small n:

n	Primitive polynomial(s)	Period
2	$1 + x + x^2$	3
3	$1 + x + x^3$; $1 + x^2 + x^3$	7
4	$1 + x + x^4$; $1 + x^3 + x^4$	15
5	$1 + x^2 + x^5$; $1 + x^3 + x^5$; ...	31
6	$1 + x + x^6$; $1 + x^5 + x^6$; ...	63
7	18 primitive polys	127
8	16 primitive polys	255
10	60 primitive polys	1023

Practical consequence

For $n = 31$: period $= 2^{31} - 1 \approx 2.1 \times 10^9$ — a very long sequence from just 31 bits of state!

State Diagram of an LFSR

Example: $n = 3$, $f(x) = 1 + x + x^3$ **(primitive)**

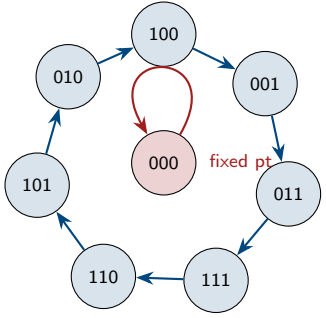
Starting from state $(1, 0, 0)$, the sequence of states is:

a_3	a_2	a_1	step	output
1	0	0	0	0
0	0	1	1	1
0	1	1	2	1
1	1	1	3	1
1	1	0	4	0
1	0	1	5	1
0	1	0	6	0
1	0	0	7	**(repeats)**

Output sequence: **0011101** | 0011101 | ...

Period $= 7 = 2^3 - 1$ ✓

State diagram (period-7 cycle + fixed point):



The all-zero state is always a fixed point (excluded from the m-sequence). A primitive polynomial gives a **single** *cycle of length* $2^n - 1$ *plus this fixed point.*

The Three Fundamental Properties of m-Sequences

Theorem (Golomb)

Every maximal-length sequence produced by an n-stage LFSR with a primitive characteristic polynomial satisfies the following three properties.

P1 — Balance

In each period of length $2^n - 1$, the number of 1s exceeds the number of 0s by exactly **one**:

$$\#\{1\text{s}\} = 2^{n-1}$$

$$\#\{0\text{s}\} = 2^{n-1} - 1$$

P2 — Run Property

Among all **runs** of consecutive identical symbols:

- $\frac{1}{2}$ of runs have length 1
- $\frac{1}{4}$ have length 2
- $\frac{1}{8}$ have length 3
- one run of 0s of length $n-1$
- one run of 1s of length n

P3 — Ideal Autocorrelation

The periodic autocorrelation function is **two-valued**:

$$C(\tau) = \begin{cases} 2^n - 1 & \tau \equiv 0 \\ -1 & \tau \not\equiv 0 \end{cases}$$

or equivalently (normalized):

$$\hat{C}(\tau) = \begin{cases} 1 & \tau = 0 \\ -\frac{1}{2^n - 1} & \tau \neq 0 \end{cases}$$

Golomb's Randomness Postulates in Detail

Why these three postulates?

A truly random infinite binary sequence has (with probability 1):

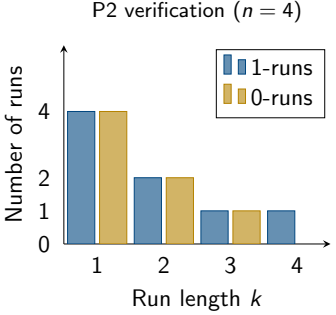
- Equal frequencies of 0s and 1s (by the Law of Large Numbers) $\rightarrow$ **P1**
- Runs of length k occurring with frequency $\approx 2^{-(k+1)} \rightarrow$ **P2**
- Autocorrelation $C(\tau) \approx 0$ for $\tau \neq 0 \rightarrow$ **P3**

A periodic sequence cannot achieve these *exactly*, but m-sequences achieve the **best possible approximation** for a given period length.

Golomb's definition (1955)

A binary sequence satisfying P1, P2, and P3 is called a **PN-sequence** (pseudonoise sequence).

Run distribution: $n = 4$, period=15



4 runs of length 1, 2 of length 2, 1 of length 3 (zeros) and 1 of length 4 (ones).

Autocorrelation of m-Sequences (P3)

Definition: For a ± 1 binary sequence $\{s_k\}$ with period $N = 2^n - 1$, the **periodic autocorrelation function (PACF)** is:

$$C(\tau) = \sum_{k=0}^{N-1} s_k \, s_{k+\tau}$$

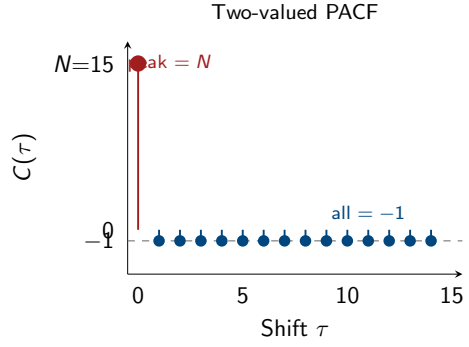
where $s_k = (-1)^{a_k}$ maps $\{0,1\} \to \{+1,-1\}$.
For an m-sequence of period $N = 2^n - 1$:

$$C(\tau) = \begin{cases} N & \text{if } \tau \equiv 0 \pmod{N} \\ -1 & \text{if } \tau \not\equiv 0 \pmod{N} \end{cases}$$

Why this matters for RNGs

The flat (nearly zero) off-peak autocorrelation means successive output values are **nearly uncorrelated** — key for a good PRNG.

PACF of an m-sequence ($n = 4$, $N = 15$):



Compare to a true random sequence where $C(\tau) \approx 0$ for $\tau \neq 0$.

The Cycle Structure Theorem

For a general (possibly non-primitive) characteristic polynomial:

If $f(x) = f_1(x)^{e_1} f_2(x)^{e_2} \cdots f_k(x)^{e_k}$ is the factorization over $\mathrm{GF}(2)$, where each $f_i(x)$ is irreducible of degree d_i and order $\mathrm{ord}(f_i) = r_i$, then the output sequences produced by the LFSR split into cycles whose lengths divide

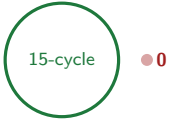
$$\mathrm{lcm}(r_1^{e_1},\ r_2^{e_2},\ \ldots,\ r_k^{e_k}).$$

The primitive case

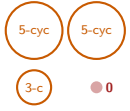
If $f(x)$ is primitive of degree n:
$\Rightarrow$ exactly **one** cycle of length $2^n - 1$
$\Rightarrow$ plus the fixed point **0**
$\Rightarrow$ total 2^n states accounted for

Comparison: primitive vs. non-primitive ($n = 4$):

Polynomial	Factorization	Cycle stru
$1 + x + x^4$	primitive	1 cycle of le
$1 + x^4$	$(1 + x)^4$	cycles of le
$1 + x + x^2 + x^3 + x^4$	reducible	mixed cy
$(1 + x + x^2)^2$	repeated	cycles div



primitive



non-primitive

Decimation of m-Sequences

Definition:
The d-**th decimation** of a sequence $\{a_k\}$ is the subsequence $\{a_{dk}\}$ (every d-th term).

Theorem (Golomb)

If $\{a_k\}$ is an m-sequence of period $N = 2^n - 1$ and $\gcd(d, N) = 1$, then the d-th decimation $\{a_{dk}\}$ is also an m-sequence of the same period.

Consequences:

- All $\phi(N)/n$ m-sequences of period N are decimations of one another — they form a single **equivalence class**
- Decimation by 2 gives a **cyclic shift**: $\{a_{2k}\}$ is just $\{a_k\}$ shifted
- This is the **shift-and-add property** of m-sequences

Shift-and-add property:

Theorem

If $\{a_k\}$ is an m-sequence and $\{b_k\}$ is any non-zero cyclic shift of $\{a_k\}$, then $a_k \oplus b_k = c_k$ where $\{c_k\}$ is *another* cyclic shift of $\{a_k\}$.

Implication for RNGs:
Two m-sequence generators running from different initial states produce sequences that XOR to yet another phase of the same sequence — a beautiful algebraic structure.

Gold sequences

Pairs of m-sequences with good cross-correlation (Gold, 1967) exploit this property for CDMA communications.

Cross-Correlation of Binary Sequences

Definition:
For two ± 1 binary sequences $\{s_k\}$ and $\{t_k\}$ of period N, the **periodic cross-correlation function (PCCF)** is:

$$C_{st}(\tau) = \sum_{k=0}^{N-1} s_k \, t_{k+\tau}$$

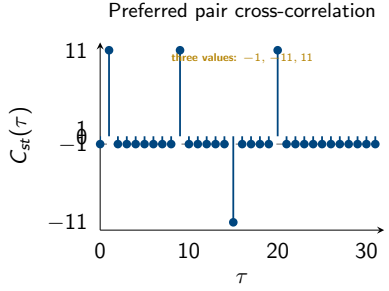
Why cross-correlation matters:

- In **CDMA**: multiple users share a channel; low cross-correlation between their codes prevents interference
- In **parallel RNG**: streams from different seeds should be uncorrelated
- In **radar/sonar**: cross-correlation used for ranging

Ideal cross-correlation

We want $|C_{st}(\tau)| \ll N$ for **all** τ when $\{s\} \neq \{t\}$.

Cross-correlation values for two m-sequences ($n = 5$):



A "preferred pair" of m-sequences achieves only **three** *cross-correlation values:* $\{-1, -(2^{(n+1)/2}+1), 2^{(n+1)/2}-1\}$ *for odd* n.

Gold Sequences and Kasami Sequences

Gold sequences (Gold, 1967):

Take a **preferred pair** of m-sequences $\{u_k\}$, $\{v_k\}$ of period $N = 2^n - 1$ (n odd or $n \equiv 2 \pmod 4$). Form:

$$g_\tau(k) = u_k \oplus v_{k+\tau}, \quad \tau = 0, 1, \ldots, N-1$$

Together with $\{u_k\}$ and $\{v_k\}$, this gives a **family of $N+2$ sequences** all of period N.

Gold's theorem

Every pair within the Gold family has cross-correlation taking only the three values $\{-1, -(2^{(n+1)/2}+1), 2^{(n+1)/2}-1\}$, achieving the **Welch lower bound** (nearly).

Summary of sequence families:

Family	Size	Max $\lvert C \rvert$	Period
m-sequences	$\phi(N)/n$	3-val	$2^n - 1$
Gold	$N+2$	$2^{(n+1)/2}+1$	$2^n - 1$
Kasami (small)	$2^{n/2}$	$2^{n/2}+1$	$2^n - 1$
Bent (even n)	large	$2^{n/2}$	varies

RNG relevance

Gold and Kasami families are used to generate **multiple independent streams** in parallel RNGs (e.g., SPRNG's LFSR generator uses Gold sequences to guarantee stream independence).

Non-Linear Feedback Shift Registers

Motivation:
Linear shift registers are **cryptographically weak**:

- Given $2n$ consecutive output bits, the Berlekamp-Massey algorithm recovers the characteristic polynomial in $O(n^2)$ time
- The sequence is *completely predictable* from short output

Non-linear alternatives:

1. **Non-linear feedback function** f — e.g., add a non-linear term to a linear recurrence
2. **Non-linear output filter** — pass LFSR output through a non-linear Boolean function (filter generator)
3. **Combination generator** — XOR outputs of multiple LFSRs through a non-linear combiner g
4. **Clock-controlled generators** — vary the clock rate of one LFSR based on the output of another (shrinking generator, alternating step generator)

Non-Linear Feedback Shift Registers (Cont.)

de Bruijn sequences

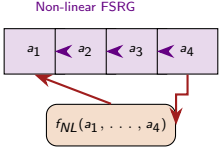
A **de Bruijn sequence of order** n is a binary sequence of period 2^n in which every n-bit pattern appears exactly once.

Produced by a **non-linear** shift register of length n.

The number of distinct de Bruijn sequences of order n is

$$2^{2^{n-1}-n}$$

For $n = 4$: $2^{8-4} = 16$ de Bruijn sequences of period 16.



LFSRs as Pseudorandom Number Generators

Practical LFSR-based PRNGs:

1. **Tausworthe generators** Combine the bits of an LFSR into w-bit words:

$$x_k = \sum_{i=0}^{w-1} a_{wk+i}\, 2^{-i-1}$$

The resulting floating-point values are uniform on $(0, 1)$.

2. **GFSR (Generalized FSR)** (Lewis & Payne, 1973) Use w-bit registers:

$$X_k = X_{k-p} \oplus X_{k-q}, \quad p > q$$

Period: $2^{wp} - 1$ if characteristic polynomial is primitive.

3. **Mersenne Twister** (Matsumoto & Nishimura, 1998)
GFSR variant with $n = 624$, period $2^{19937} - 1$ — the most widely used PRNG today.

LFSRs as Pseudorandom Number Generators (Cont.)

Comparison of LFSR-based generators:

Generator	Period	State bits
Basic LFSR	$2^n - 1$	n
Tausworthe	$2^n - 1$	n
GFSR	$2^{wp} - 1$	wp
Mersenne Twister	$2^{19937} - 1$	19937
Combined Tausworthe	$\approx 2^{88}$	88

Key advantage over LCGs

LFSR-based generators:

- No lattice structure in high dimensions
- Excellent autocorrelation (P3)
- Very fast in hardware (XOR gates only)
- Long provable periods

The Berlekamp-Massey Algorithm

Problem: Given a finite binary sequence $s_0, s_1, \ldots, s_{N-1}$, find the **shortest LFSR** that generates it.

The linear complexity $L(s)$: the length of the shortest LFSR generating $\{s_k\}$.

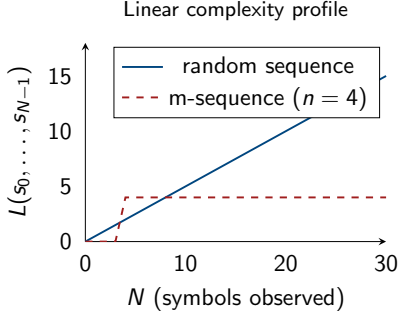
Berlekamp-Massey algorithm

Finds $f(x)$ (characteristic polynomial of length $L(s)$) from $\{s_k\}$ in $O(N^2)$ time — or $O(N \log N)$ with improvements.

Requires only $2L(s) \leq N$ output symbols to uniquely determine the minimal LFSR.

Cryptographic implication:
An m-sequence of length $2^n - 1$ has $L = n$ — catastrophically low! A stream cipher must have $L \gg n$ to resist this attack.

Linear complexity profile:



A random sequence has $L \approx N/2$ (unpredictable). An m-sequence has $L = n$ (predictable after $2n$ bits).

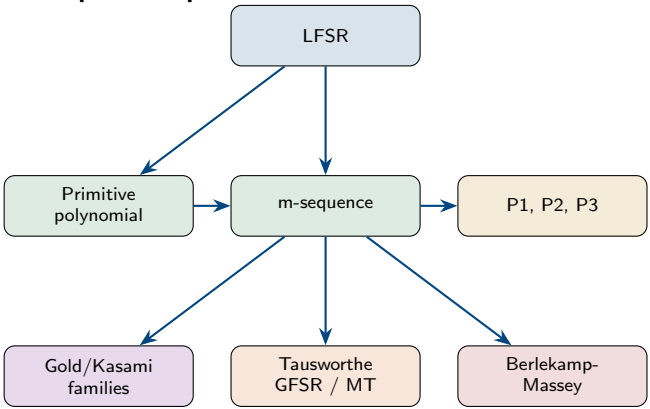
Key Results Summary

What we covered (Golomb, 1982):

Topic	Key Result
LFSR definition	$a_{n+k} = \bigoplus c_i a_{k+n-i}$
Primitive polynomial	Yields m-sequence of period $2^n - 1$
# Primitive polys	$\phi(2^n - 1)/n$
P1: Balance	2^{n-1} ones, $2^{n-1} - 1$ zeros
P2: Run property	2^{n-k-1} runs of length k
P3: Autocorrelation	$C(\tau) = N$ or -1
Shift-and-add	$a_k \oplus a_{k+\tau} = a_{k+\sigma}$
Decimation	$\gcd(d, N) = 1 \Rightarrow$ m-sequence
Gold sequences	$N + 2$ seqs, $\lvert C_{st}\rvert \leq 2^{(n+1)/2} + 1$
Linear complexity	$L = n$ (cryptographically weak)
Berlekamp-Massey	Recovers LFSR from $2n$ bits
Mersenne Twister	GFSR, period $2^{19937} - 1$

Key Results Summary (Cont.)

Conceptual map:



Discussion Questions

1. A 4-stage LFSR with characteristic polynomial $f(x) = 1 + x^2 + x^4$ is *not* primitive. Factor $f(x)$ over $\mathrm{GF}(2)$ and describe the resulting cycle structure. What is the maximum period achievable?

2. Verify Golomb's three randomness postulates P1, P2, P3 for the m-sequence generated by $f(x) = 1 + x + x^4$ with initial state $(1, 0, 0, 0)$. Write out the full period-15 sequence.

3. The Berlekamp-Massey algorithm can break any LFSR-based generator from $2n$ bits of output. Explain why the Mersenne Twister (despite using an LFSR structure) is still considered secure enough for simulation (though not for cryptography).

4. Explain the **shift-and-add property** of m-sequences geometrically. Why does this property make m-sequences attractive for generating *multiple independent streams* in parallel Monte Carlo simulations?

5. Compare and contrast LCGs (Chapter 3 of H&H) and LFSRs: period, statistical quality, lattice structure, hardware efficiency, and predictability. When would you prefer each?

Further Reading & References

- S. W. Golomb, *Shift Register Sequences* (Revised Edition). Aegean Park Press, Laguna Hills, CA, 1982.
- R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. Inform. Theory*, 13(4):619–621, 1967.
- E. R. Berlekamp, *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.
- J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, 15(1):122–127, 1969.
- M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.
- J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. Methuen, London, 1964.
- D. E. Knuth, *The Art of Computer Programming*, Vol. 2: Seminumerical Algorithms, 3rd ed. Addison-Wesley, 1997. (§3.2.2: Linear Feedback Shift Registers)